DESCRIPTION

A BIT STREAM BUFFERING AND DEMULTIPLEXING APPARATUS FOR A DVD AUDIO DECODING SYSTEM

TECHNICAL FIELD

This invention relates to the implementation of a data buffering and demultiplexing apparatus for a DVD-Audio decoder system.

BACKGROUND ART

“DVD Specifications for Read-Only Disc Part 4 Audio Specifications Version 0.9”, referred to as "DVD Audio specifications" hereinafter, specifies a new type of data stream, an audio still video program stream. The audio still video stream is not multiplexed with the audio program stream but stored as separate object on its own. This is different from “DVD Specifications for Read-Only Part 3 Video Specifications Version 1.1”, referred to as "DVD Video specifications" hereinafter, where all elementary data streams such as audio, video and sub-picture are multiplexed into one logical program stream. An audio still video object (ASVOB) is formed from three elementary stream objects, namely, highlight information, 1 to 3 sub-pictures and still video. An alternate form exists where an audio still video is formed only by a still

video object. A collection of audio still video objects makes up an audio still video unit (ASVU). A limited number of audio still video objects can exist in one audio still video unit. According to DVD Audio Specifications, one audio still video unit is limited to 99 audio still video objects, and the size is limited to 2 Megabytes. A collection of audio still video units makes up an audio still video stream (ASVUS).

In the DVD Audio specifications, a DVD audio decoder must buffer the whole audio still video unit in an audio still video unit buffer. Two demultiplexers capable of decoding program streams are needed. One of them demultiplexes the audio still video unit program stream from an audio still video unit buffer, and the other demultiplexes the audio program stream from a DVD Audio disc. In addition, DVD Audio specifications also stipulate that the audio still video objects in an audio still video unit can be accessed in any unknown order until the audio still video is to be presently selected. Thus, the starting location of each audio still video in an audio still video unit needs to be known.

Figure 1 shows an example of an implementation based on the decoder model specified in the DVD Audio specifications. When input data is a type of audio still video program stream, the input data is directed by a

selector to be stored (or buffered) into an audio still video buffer (ASVU Buffer) via a pre-loading terminal 0. When the buffering of the data is completed, the selector is switched back to a decoding position 1. When input data is a type of audio program stream, the data is directed to an audio program stream demultiplexer, DEMUX2 and then the demultiplexed data, such as the audio elementary stream, is written into an audio bit buffer and other buffers such as a real time text bit buffer. At the same time, the data from the audio still video unit buffer (ASVU Buffer) is read into the other demultiplexer, DEMUX1, which demultiplexes the data to be written into video, sub-picture and highlight bit buffers. An audio still video address (ASV Address) table stores the start and/or end address of each audio still video object in the audio still video unit. These addresses are used to select the correct audio still video object to be sent to DEMUX1.

Looking at Figure 1, it is obvious that the incoming data stream comprises two independent multiplexed program streams, namely, audio still video program stream and audio program stream. From a DVD Audio decoder point of view, this is different from that of a DVD Video decoder. Therefore, two program stream demultiplexers are needed for both the audio still video program and audio program. This solution is more costly, because the presently existing DVD

Video decoder system only requires one demultiplexer. Alternatively, a single high speed demultiplexer that could demultiplex two streams simultaneously is needed. This would require a new demultiplexer that is capable of decoding at a rate two times that of a conventional demultiplexer. Again, this is more costly than using a slower speed demultiplexer that already exists in the DVD Video decoder.

This invention discloses a method that buffers the demultiplexed audio still video unit stream after it has been demultiplexed by a program stream demultiplexer similar to that used in the current DVD Video decoder system. This means that the invention can be implemented by effectively using only one program stream demultiplexer.

In addition, the size of the bit buffers in the system for storing demultiplexed elementary stream can be reduced. This saving comes from the fact that the size of the audio still video unit after it has been demultiplexed is smaller in size than the original program stream. The other saving comes from the fact that a separate video bit buffer, sub-picture bit buffer and highlight bit buffer are no longer required. The audio still video unit bit buffer which stores the demultiplexed elementary audio still video data is already in the bit buffer format. This also improves the time it takes to access a specific audio still

video object. It is no longer necessary to send the audio still video object to a demultiplexer first.

In a conventional system, such as a DVD Video demultiplexing of audio/video stream, the program stream is demultiplexed only when it is needed. By performing demultiplexing early during pre-loading of audio still video unit program stream into the audio still video unit bit buffer, the system can detect a potential bit stream syntax error in advance, before the data is decoded.

SUMMARY OF THE INVENTION

For the purpose of solving the above-described problems, the bit stream buffering and demultiplexing architecture according to the present invention was designed.

In order to keep the cost of the DVD Audio decoder down by not adding an additional demultiplexer in the system, a buffering method which demultiplexes the audio still video unit program stream during the pre-loading to the audio still video unit bit buffer is invented. In order to reduce the amount of bit-stream buffers used in a decoder system, a means for bit buffer memory sharing is invented. In order to help the decoder system better manages bit-stream errors, error checking the program stream syntax during demultiplexing allow the

decoder to detect a stream error early, before the DVD Audio decoder needs to present any data to the user. In order to speed up access time in accessing a particular audio still video object from the audio still video unit bit buffer, a demultiplexed audio still video unit program stream is stored in the audio still video unit bit buffer. The address locations of each object in an audio still video unit are easily available to speed up accesses as well.

According to an essential feature of the present invention, a bit stream buffering and demultiplexing apparatus for a DVD Audio decoding system comprises: a demultiplexer for demultiplexing a coded program stream to elementary streams; an audio still video unit bit buffer for storing demultiplexed audio still video program streams; a bit stream buffer for storing a demultiplexed audio program stream, and an audio still video object address pointer table storing address locations of the demultiplexed audio still video program streams.

In this construction, the audio still video object address pointer table may further store status information of the demultiplexed audio still video program streams.

Also, the demultiplexer may comprise: a means for demultiplexing the coded program stream to

elementary streams, and a means for switching of writing to the audio still video unit bit buffer from the bit stream buffers, the switching occurring whenever the input bitstream is an audio still video program stream.

Also, the audio still video unit bit buffer for storing the demultiplexed audio still video bit streams may comprise: a means for storing elementary streams of audio still video, and a means for storing start address pointers of all or a sub group of elementary streams of an audio still video unit.

In this construction, the audio still video unit bit buffer further may comprise a means for storing status information relating to all or sub group of an audio still video unit.

Also, the audio still video address pointer table may comprise: a means for storing start and/or end address pointers of all or a sub group of elementary streams of an audio still video unit, and a means for storing status information relating to all or a sub group of an audio still video unit.

Moreover, the status information storing means may comprise: a means for storing syntax error information, and a means for storing other information related to the audio still video unit.

Another aspect of the present invention provides

a bit stream buffering and demultiplexing method for a DVD Audio decoding system, which comprises the steps of: demultiplexing a coded program stream to elementary streams; storing a demultiplexed audio still video program streams; storing demultiplexed audio program stream, and storing address locations of the demultiplexed audio still video program streams, wherein the demultiplexing step includes a step of demultiplexing the audio still video unit program stream during a pre-loading to the audio still video unit bit buffer.

Further another aspect of the present invention provides a DVD Audio decoding system having a bit stream buffer and a demultiplexer, wherein the multiplexer is only one demultiplexer which generates an audio still video address pointer table indicating an access address for each audio still video object, to demultiplex both the audio program stream and the audio still video program stream, and the bit stream buffer comprises means for storing demultiplexed audio still video data in an elementary format.

The DVD Audio decoder system reads in a bit stream from the DVD Audio disc and sends it to the demultiplexer. For the DVD Audio decoder, the audio still video unit program stream is read from the disc first and passes to the demultiplexer. The demultiplexer strips off

the program stream layer and stores elementary video, highlight information and sub-picture streams in the audio still video unit bit buffer. This is done during the audio still video unit pre-loading specified in the DVD Audio specifications. The demultiplexer also checks the structure of the audio still video program stream to make sure it conforms to the structure outlined in DVD Audio specifications. Bit stream errors are reported to the system. The demultiplexer also keeps track of the location of each audio still video objects demultiplexed. These address locations are buffered to allow random access to a specific audio still video object during audio program decoding. After the decoder completes the pre-loading process, audio program stream is read from the DVD Audio disc. The same demultiplexer then demultiplexes the audio program stream that contains audio and other optional streams, such as real-time text. Demultiplexed elementary data are stored in appropriate bit buffers.

From the audio bit buffers, the audio decoder reads the audio elementary stream, decodes and presents the data out. At the same time, using the audio still video objects address stored in the pointer table, the video, sub-picture and highlight information decoders read in the appropriate audio still video object, decode, and present the data to the user. The presentation order of the audio

still video objects depends on presentation information stored in the DVD Audio disc or from the interactive controls of the DVD Audio decoder system user.

BRIEF DESCRIPTION OF DRAWINGS

These and other objects and features of the present invention will be readily understood from the following detailed description taken in conjunction with preferred embodiments thereof with reference to the accompanying drawings, in which like parts are designated by like reference numerals and in which:

Figure 1 is a prior-art of the current invention;

Figure 2 is an example embodiment of the invented DVD Audio stream buffering and demultiplexing system;

Figure 3 is an example configuration of the audio still video object address pointer table and audio still video unit bit buffer mapping of the embodiment of Figure 2; and

Figure 4 is another example configuration of the audio still video object address pointer table and audio still video unit bit buffer mapping of the embodiment of Figure 2.

DETAILED DESCRIPTION OF THE INVENTION

Before the description proceeds, it is to be noted

that, since the basic structures of the preferred embodiments are similar, like parts are designated by the same reference numerals throughout the accompanying drawings.

An example of an embodiment of the present invention is described with reference to Figure 2. In Figure 2, a program stream comes in from a program stream input terminal 100 to a demultiplexer, DEMUX 101. The input stream is multiplexed according to ISO13818-1 MPEG-2 Program Stream Standard as well as to DVD Audio and Video specifications. DEMUX 101 demultiplexes the program stream into elementary data streams. For the current embodiment of the present invention, but not limited by this, DEMUX supports demultiplexing into the following elementary streams: video, sub-picture, highlight information, audio and other data, such as real-time text. The DEMUX 101 demultiplexes the input stream and then the demultiplexed elementary data streams are written into video bit buffers 107, sub-picture bit buffers 108, highlight bit buffers 109, audio bit buffer 110 and other buffers 111, such as a real-time text bit buffer.

In this embodiment, video buffers are logical buffers that store all the video objects of all the audio still video objects contained in an audio still video unit. The same is said for the sub-picture bit buffers 108 and the highlight bit buffers 109. These 3 groups of bit

buffers make up the audio still video unit bit buffer 300. More details on the mapping of this buffer shall be stated later.

There are two types of multiplexed program streams input from the input terminal 100 to DEMUX 101. An audio still video unit stream is a multiplexed stream of video, sub-picture and highlight data. An audio program stream is a multiplexed stream of audio and real-time text data. Accordingly, DEMUX 101 may include a switch means for switching the writing of the demultiplexed program streams between the audio still video unit bit buffer (300) and the bit stream buffers (110, 111) in accordance with the types of the input program streams. A selector as shown in Figure 1 may be used as a switch means. Thus, when the input data is a type of audio still video program stream, the demultiplexed data output of DEMUX 101 is directed by the selector to be stored in the audio still video unit bit buffer (300). When the input data is a type of audio program stream, the demultiplexed data output of DEMUX 101 is directed to the bit stream buffers (110, 111).

When the system is performing audio still video pre-loading, the audio still video program stream is inputted to DEMUX. DEMUX writes the demultiplexed data via buses 102, 103 and 104 into the respective bit buffers 107, 108 and 109 in the audio still video unit buffer 300. This

unit buffer is similar to the ASVU buffer stated in the prior art shown in Fig. 1 except that the elementary data streams are stored instead. During the demultiplexing of audio still video program stream, DEMUX also calculates the start and end location of each of the video, sub-picture and highlight elementary streams and stores these addresses in an audio still video object address pointer table 200. This table is essential for random accessing of audio still video object during the decoding phase (or mode) of the decoder.

During audio still video unit demultiplexing, DEMUX can perform various types of stream integrity checks, such as program stream syntax check or audio still video stream structure check. The number of audio still video objects can be counted and then confirmed with the number stored elsewhere on the disc. The order of video, sub-picture and highlight in an audio still video object can also be double-checked to confirm the validity of the stream. The size of the audio still video unit can also be confirmed against the limit set by the Specification. All this information can provide a good indication to the decoder as to the data integrity of the disc.

When the system completes audio still video pre-loading, the system inputs the audio program stream to start audio decoding. During this time, DEMUX

demultiplexes the audio program stream into audio and real-time text elementary streams, and stores the elementary streams data into their respective bit buffers 110 and 111 via buses 105 and 106. At this time, the video, sub-picture, highlight, audio and real-time text elementary streams are read from their respective bit buffers simultaneously and sent to their respective decoder for decoding. The video, sub-picture and highlight elementary streams are accessed depending on which audio still video object within the audio still video unit has been selected for decoding. This information may not be known until 0.4 second before the audio still video object is to be presented, according to DVD Audio Specifications. The audio still video object address pointer table 200 stores the information needed by the decoder to read the correct data from the audio still video unit bit buffer 300.

Figure 3 shows an embodiment of the audio still video object address pointer table 200 and the audio still video unit bit buffer 300. In this embodiment, DEMUX stores the start address pointer of each audio still video object (ASVOB 1-99) it encounters when demultiplexing the audio still video unit program stream as in the audio still video object address pointer table 200. Each of the start addresses in turn points to a start position of each of the audio still video objects stored in the audio still video

unit bit buffer 300. The beginning portion of each audio still video object in the audio still video bit buffer further contains pointer addresses that point to the start of sub-picture bit buffer and video bit buffer for that particular audio still video object. Highlight bit buffer does not need pointer address as it immediately follows the video pointer addresses and status information data. It is noted here that the audio still video address pointer table (200) may store start and/or end address pointers of all or a sub group of elementary streams of an audio still video unit.

Referring to Figure 3, as to an audio still video object 1 (ASVOB1), an address pointer 201 in the address pointer table 200 points to the beginning of the audio still video object 1 (ASVOB1) in the audio still video bit buffer 300. An arrow line 202 indicates this pointer in Figure 3. The sub-picture address pointer 203 for ASVOB1 in turn points to a start location of a sub-picture bit buffer 206 for ASVOB1, and an arrow line 208 shows this pointer. A video address pointer 204 for ASVOB1 immediately after the sub-picture address pointer 203 points to a start location of a video bit buffer 207 for ASVOB1, and an arrow line 209 shows this pointer. Immediately after the video address pointer 204 for ASVOB1, extra status information of ASVOB1 is stored that is

indicative of, such as, whether the current audio still video object contains valid highlight data, or syntax error information. Numeral 301 shows the status information for ASVOB1. A highlight bit buffer 205 for ASVOB1 follows immediately after the status information 301 for ASVOB1. In cases when no highlight data or sub-picture exist, setting the sub-picture pointer address to 0 will indicate that only a video bit buffer exists in the bit stream.

For most implementation, the audio still video address pointer table would be implemented using an internal static random access memory. For the audio still video unit bit buffer, due to its larger size, it is usually implemented as part of a system memory in an external dynamic random access memory. This particular embodiment for the audio still video address pointer table allows part of the address pointer to be stored in the cheaper dynamic random access memory typically used for the audio still video unit bit buffer. The trade off for such a system would be a longer time to access the addresses to the audio still video objects.

Figure 4 shows an alternate embodiment of the audio still video object address pointer table 200 and audio still video unit bit buffer 300, related to the current invention. In this embodiment, the audio still video object address pointer table contains the address

pointers needed to access each of the audio still video objects in the audio still video unit bit buffer. The table also contains extra status information of each audio still video object to store syntax error information and extra status data.

Unlike previous embodiment in Figure 3, all pointer addresses to access the audio still video objects are stored in the pointer table 200. This embodiment has an advantage of faster accesses to the audio still video object start address, with the tradeoff of a larger pointer table. The audio still video object start address points to the start address of the specific audio still video object in the audio still video unit buffer. This also points to the highlight bit buffer of the specified audio still video object (ASVOB). The video address pointer points to the specific video bit buffer of the specified audio still video object in the audio still video bit buffer. The start address of each sub-picture bit buffer is calculated indirectly from the audio still video object address pointer.

In this embodiment, size of the highlight bit buffer is limited to 704bytes. Accordingly, the start address of sub-picture bit buffer is 704bytes offset from the start of highlight bit buffer. In the case when no valid highlight information exists in the bit buffer, the

status information field for the specific audio still video object will indicate such a condition and video address pointer will have a value equal to highlight bit buffer start address offset by 704bytes.

The decoder uses the address information stored in the audio still video address pointer table and/or the audio still video unit bit buffer to access the correct audio still video object bit buffers quickly. This is very important for implementing fast random access functions for audio still videos.

Although the present invention has been described in connection with the preferred embodiments thereof with reference to the accompanying drawings, it is to be noted that various changes and modifications will be apparent to those skilled in the art. Such changes and modifications are to be understood as included within the scope of the present invention as defined by the appended claims, unless they depart therefrom.

INDUSTRIAL APPLICABILITY

The effect of this invention is a cost efficient implementation of a bit stream buffering and demultiplexing system for DVD Audio decoder system. This is due to the use of only one demultiplexer. Storing of audio still video unit in an elementary form also has the advantages of

reducing the size of bit buffer memories and speeding up access to the audio still video object data.